REDES NEURAIS COM TENSORFLOW

CLASSIFICAÇÃO: REDE NEURAL FEEDFORWARD

DIEGO RODRIGUES DSC

INFNET

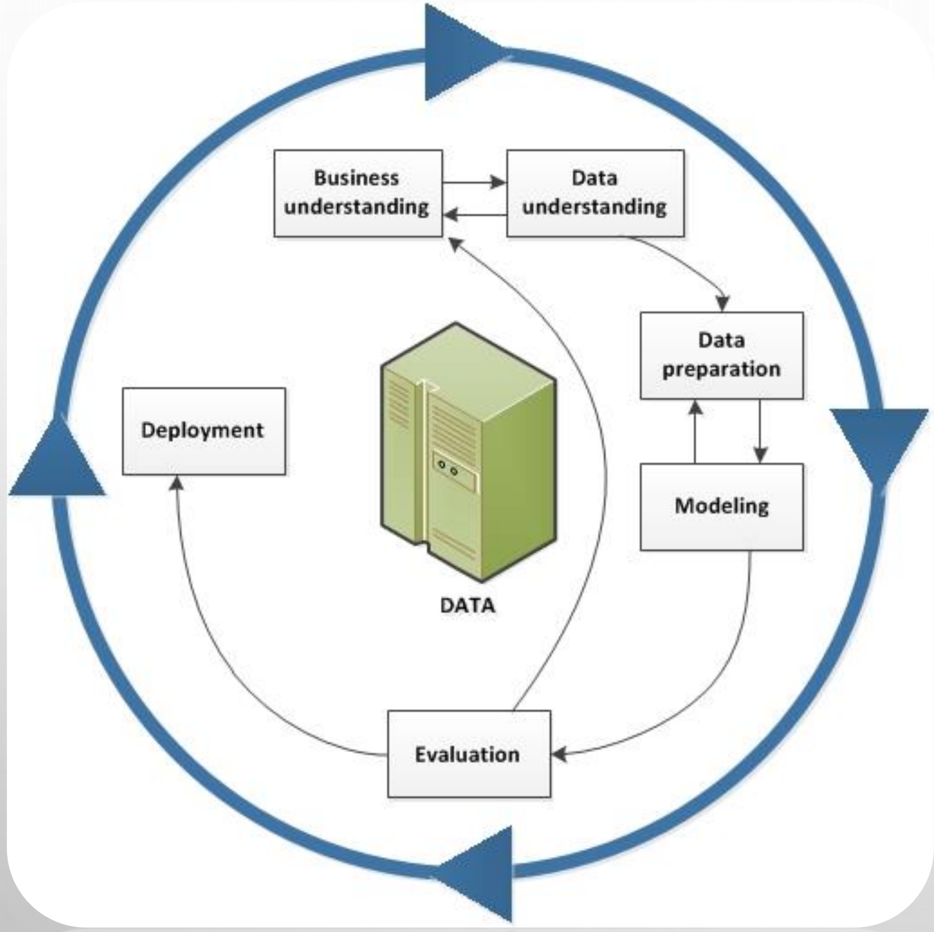
CRONOGRAMA

Dia	Aula	Trab
29/07	Perceptron de Rosenblatt	
31/07	Classificação: Neurônio Sigmóide	
05/08	Classificação: Rede Neural Feedforward	Grupos
07/08	Classificação: Treinamento Robusto	
12/08	Regressão	Base de Dados
14/08	Agrupamento	
19/08	Séries Temporais	Modelos
21/08	Apresentação dos Trabalhos Parte I	

CLASSIFICAÇÃO : REDE NEURAL FEEDFORWARD

- PARTE 1 : META HEURÍSTICA DE TREINAMENTO
  - BUSINESS UNDERSTANDING
  - DATA UNDERSTANDING & PREPARATION
  - MODELAGEM
  - VALIDAÇÃO
- PARTE 2 : PRÁTICA
  - NOTEBOOK: CLASSIFICADOR IRIS “HALF” / “FULL” LEARNING
- PARTE 3 : TRABALHOS
  - ESCOPO & EVOLUÇÃO

PARTE 1 : TEORIA



CROSS INDUSTRY PROCESS FOR DATA MINING (CRISP-DM)

BUSINESS UNDERSTANDING

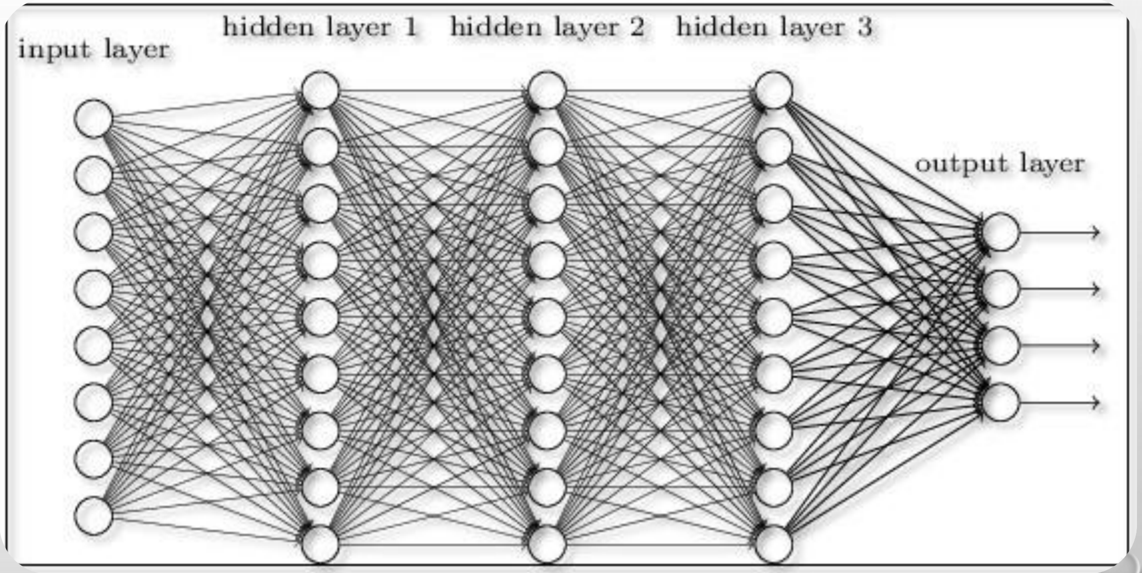
NOVO CICLO CRISP

Algoritmo	Representação	Preparação	Modelagem	Validação
• Reta 2 Pontos • NN 10% VAL • NN 10 Folds	• 2D • 2D • 2D	• Nenhuma • Nenhuma • Scale	• Reta 2 Pontos • NN Básica • NN Hidden	• Nenhuma • A/P/R • A/P/R

- Garantir estabilidade no treinamento
- Chegar a mesma solução independente do experimento
- Garantir Generalização

ANÁLISE DE NEGÓCIO

- Reprodutibilidade do Experimento
  - Controlar SEED do Numpy & Keras.
  - Mitigar o efeito da inicialização dos Parâmetros



DATA UNDERSTANDING & PREPARATION

DATA PREPARATION

Quantificação dos Atributos

- Transformar todos os atributos em atributos numéricos.

Normalização

- Transformar todos os atributos para a mesma faixa dinâmica, de maneira a assegurar que todos tenham o mesmo "peso numérico" para o treinamento do modelo.

Seleção de Atributos

- Escolher os atributos que mais impactem no resultado do modelo.

Extração de Atributos

- Transformar o Espaço de Atributos para facilitar a resolução do problema.

ATRIBUTOS CATEGÓRICOS

One Hot Encoding

Gender
Female
Male
Male
Female

Gender
1
0
0
1

Food Name	Categorical #	Calories
Apple	1	95
Chicken	2	231
Broccoli	3	50

→

Apple	Chicken	Broccoli	Calories
1	0	0	95
0	1	0	231
0	0	1	50

DATAS

Componentes da Data

- Ano
- Mês
- Dia
- Dia do Ano
- Dia da Semana
- Hora
- Minuto
- Segundo

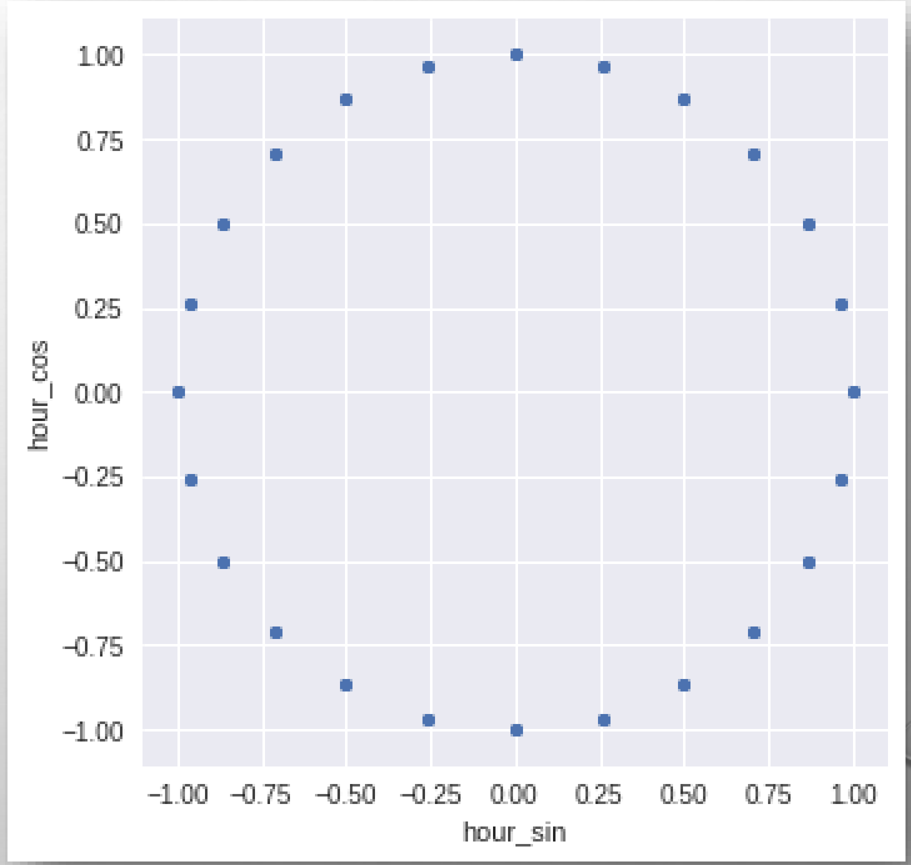
Flags

- É final de semana
- É feriado

Diferença entre Datas

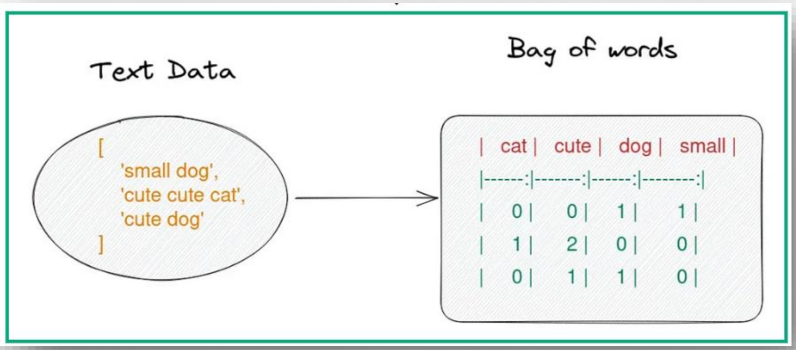
- Diferença em Dias
- Diferença em Horas
- Diferença em Meses

Encoding Cíclico



ATRIBUTOS TEXTUAIS

BAG OF WORDS



TF-IDF

Variants of term frequency (tf) weight

weighting scheme	tf weight
binary	$0, 1$
raw count	$f_{t,d}$
term frequency	$f_{t,d} \Big/ \sum_{t' \in d} f_{t',d}$
log normalization	$\log(1 + f_{t,d})$
double normalization 0.5	$0.5 + 0.5 \cdot \frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$
double normalization K	$K + (1 - K)\frac{f_{t,d}}{\max_{\{t' \in d\}} f_{t',d}}$

Variants of inverse document frequency (idf) weight

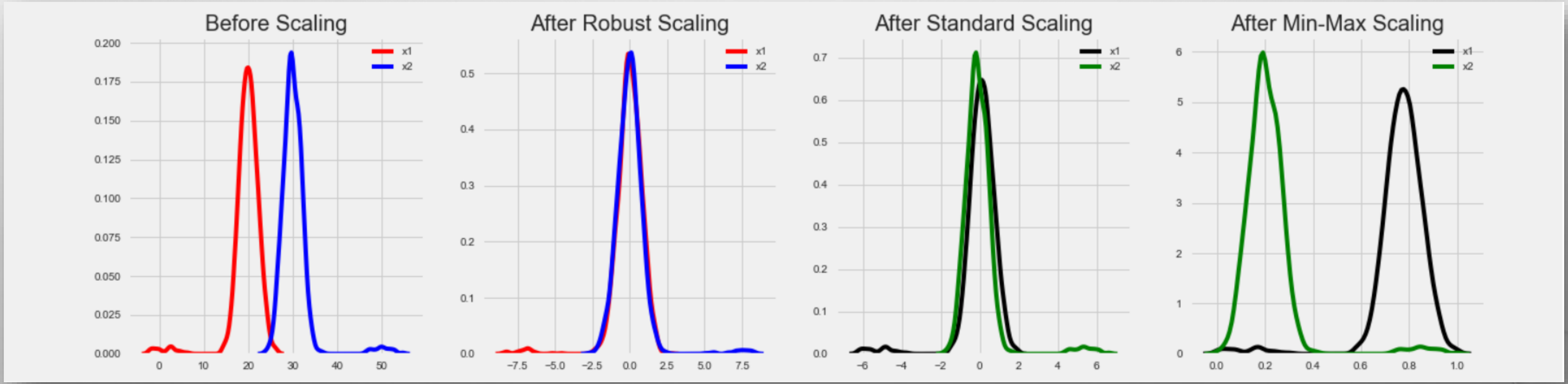
weighting scheme	idf weight ($n_t = \|\{d \in D : t \in d\}\|$)
unary	1
inverse document frequency	$\log \frac{N}{n_t} = -\log \frac{n_t}{N}$
inverse document frequency smooth	$\log\left(\frac{N}{1 + n_t}\right) + 1$
inverse document frequency max	$\log\left(\frac{\max_{\{t' \in d\}} n_{t'}}{1 + n_t}\right)$
probabilistic inverse document frequency	$\log \frac{N - n_t}{n_t}$

TF-IDF Calculation Example

Words	Count		Term Frequency (TF)		Inverse Document Frequency (IDF)	TF * IDF	
	Document 1	Document 2	Document 1	Document 2		Document 1	Document 2
read	1	1	0.17	0.17	0	0	0
svm	1	0	0.17	0	0.3	**0.05**	0
algorithm	1	1	0.17	0.17	0	0	0
article	1	1	0.17	0.17	0	0	0
dataaspirant	1	1	0.17	0.17	0	0	0
blog	1	1	0.17	0.17	0	0	0
randomforest	0	1	0	0.17	0.3	0	**0.05**

NORMALIZAÇÃO

- Garantir que as variáveis possuam a mesma escala
- Mesmo efeito numérico na otimização independente da escala.
- Transformar de outra distribuição para distribuição normal



TÉCNICAS DE SELEÇÃO DE ATRIBUTOS

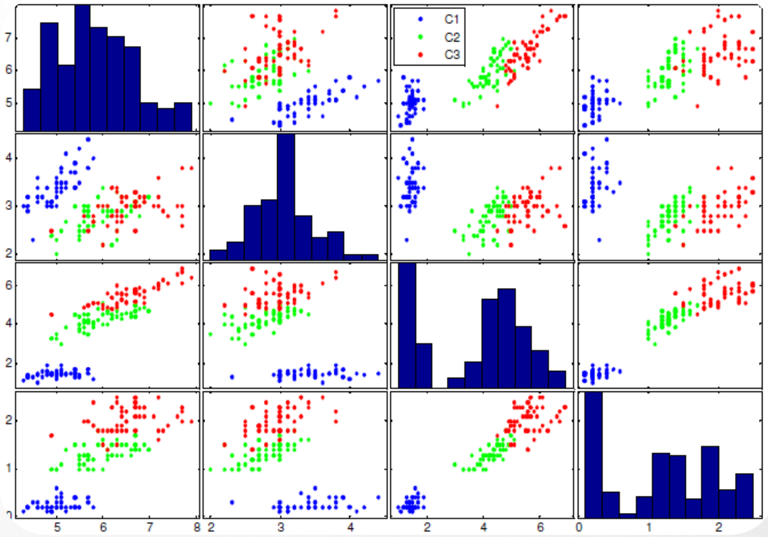
Filtragem – mede a relação entre atributos ou atributos e classes, utilizando estatísticas, sem depender do modelo.

- **Coeficiente de Correlação de Pearson –** Estatística que mede a relação linear entre duas variáveis aleatórias.
- **Teste T de diferença de médias –** Informa se a média de um determinado atributo muda de acordo com uma categoria binária.
- **ANOVA –** O mesmo que o teste T, mas serve para múltiplas categoria.
- **Informação Mútua –** Estatística que mede relação não-linear entre duas variáveis aleatórias.

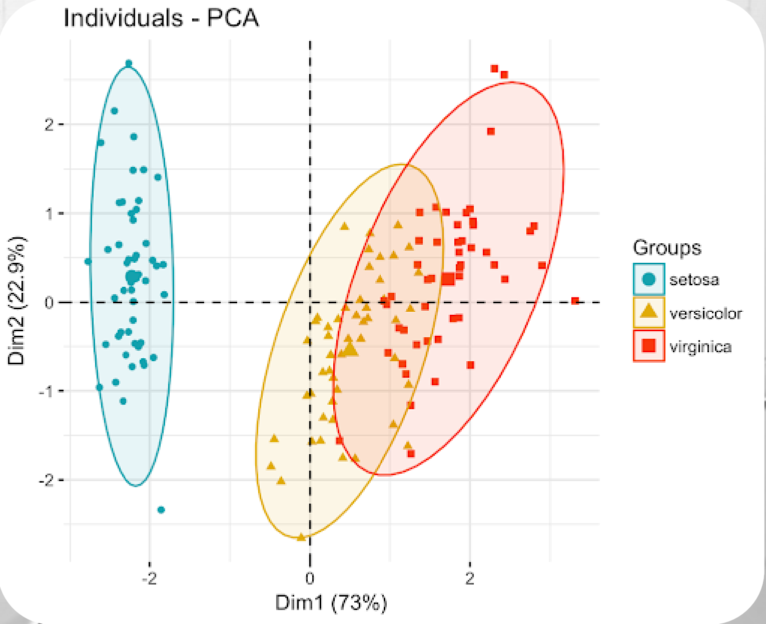
Wrapper – mede a relação entre atributos e classes, utilizando um modelo treinado.

- **Gini –** Estatística que representa a importância de um atributo na divisão da base de dados por uma árvore de decisão.
- **Relevância –** Estatística que representa a variação causada na saída do modelo quando um atributo é substituído por sua média.

EXTRAÇÃO DE ATRIBUTOS ANÁLISE DE COMPONENTES PRINCIPAIS (PCA)

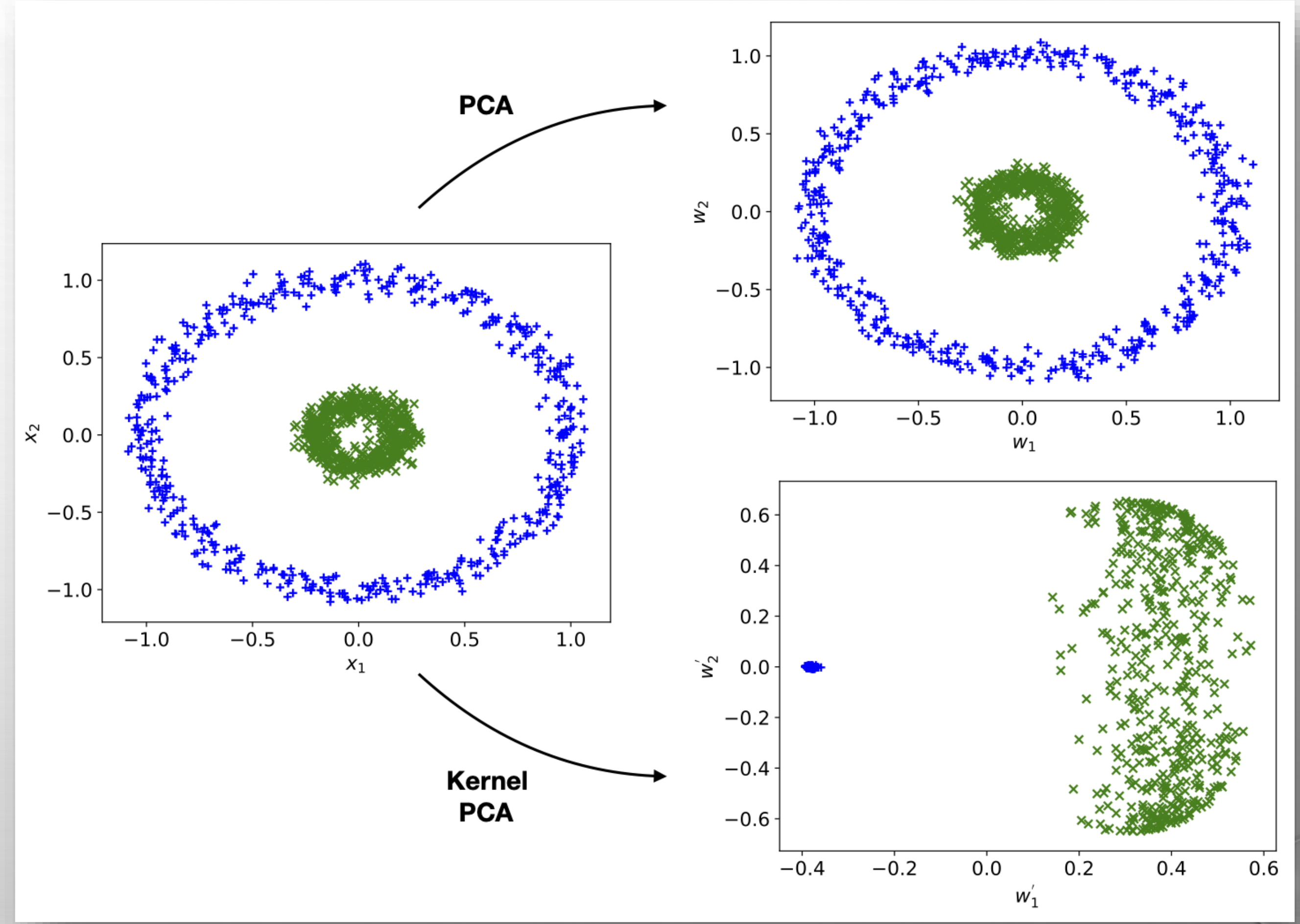


- Garantir que as variáveis independentes sejam descorrelacionadas.
- Identificar novas direções com maior concentração de energia / informação.
- Variáveis transformadas perdem o sentido físico.



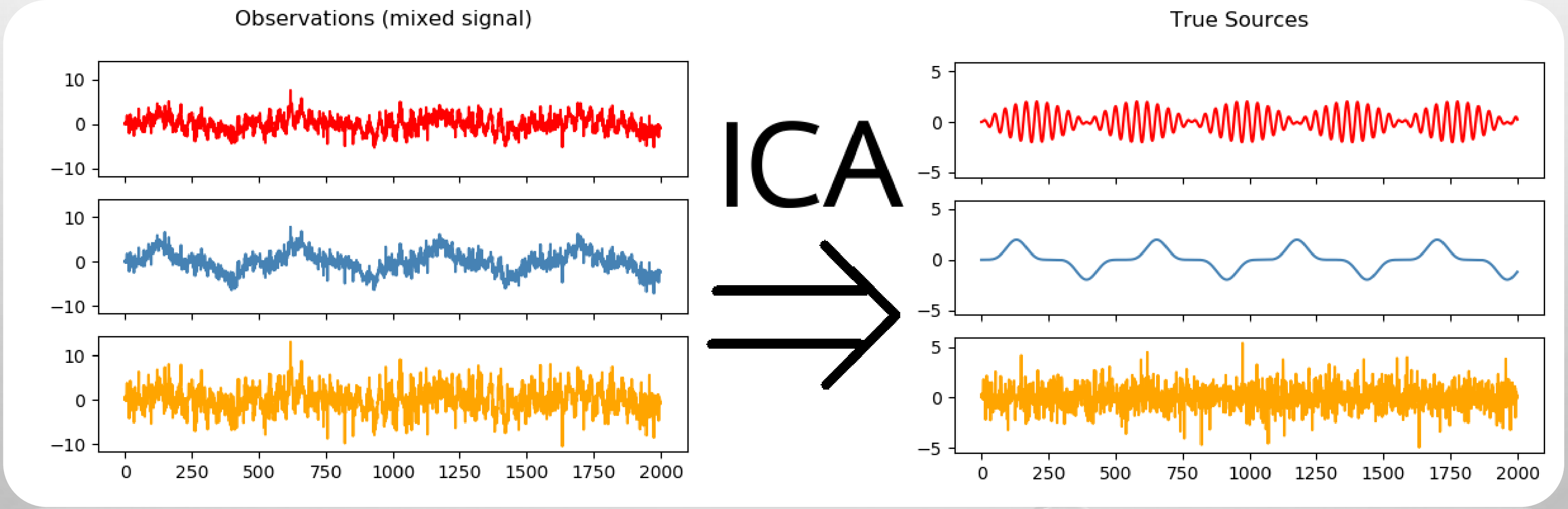
EXTRAÇÃO DE ATRIBUTOS – KERNEL PCA

- Identifica novo espaço que favoreça a modelagem.
- Como selecionar o Kernel Adequado?



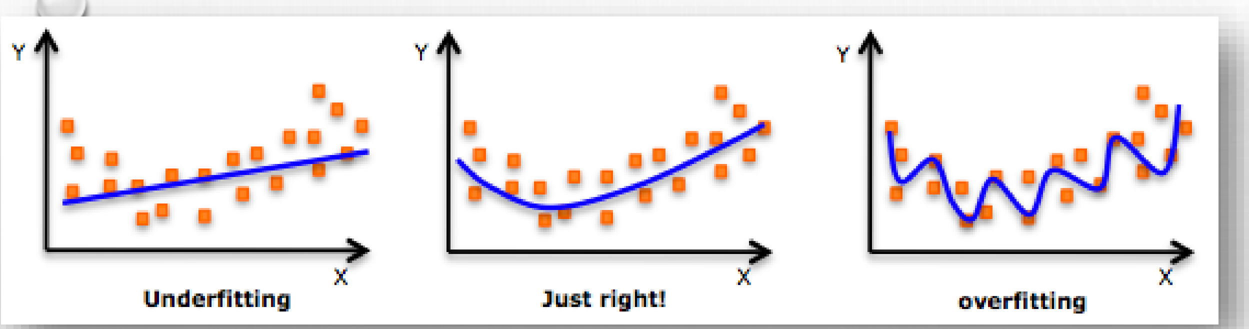
ANÁLISE DE COMPONENTES INDEPENDENTES

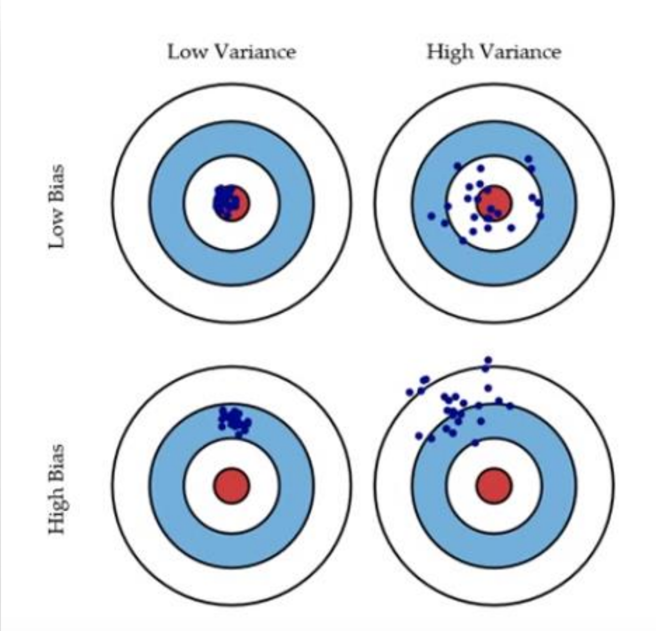
- Garantir que as variáveis independentes sejam independentes
- Identificar principais direções de não-gaussianidade

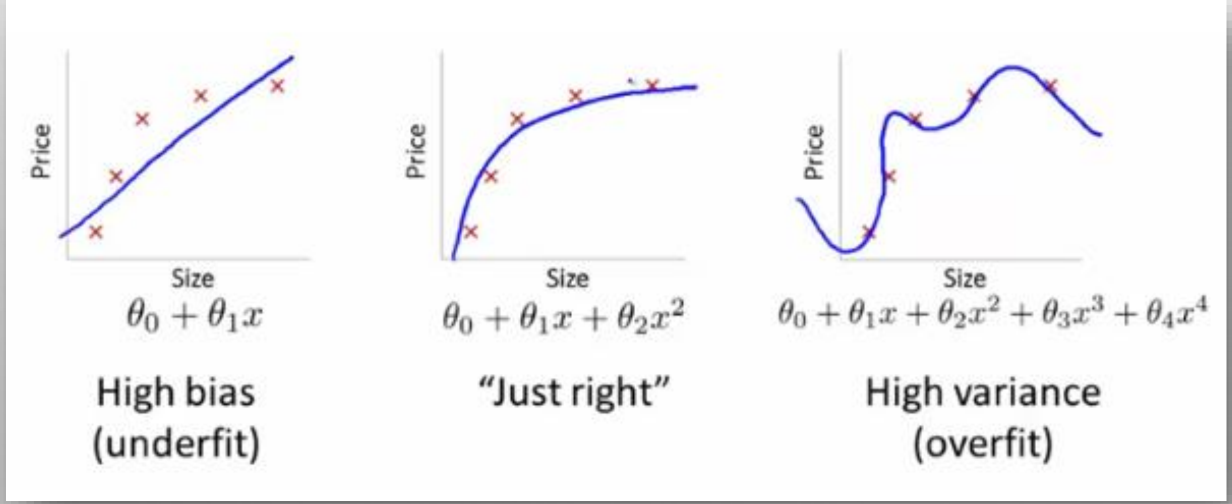


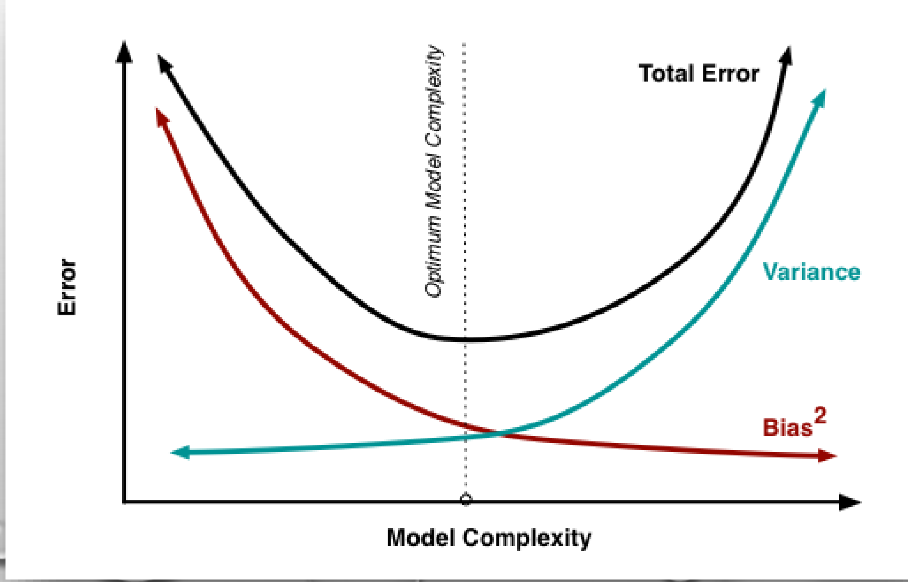
MODELING

BIAS x VARIANCE









REGULARIZAÇÃO

In mathematics, statistics, finance,[1] and computer science, particularly in machine learning and inverse problems, **regularization** is a process that changes the result answer to be "simpler". It is often used to obtain results for ill-posed problems or to prevent overfitting.[2]

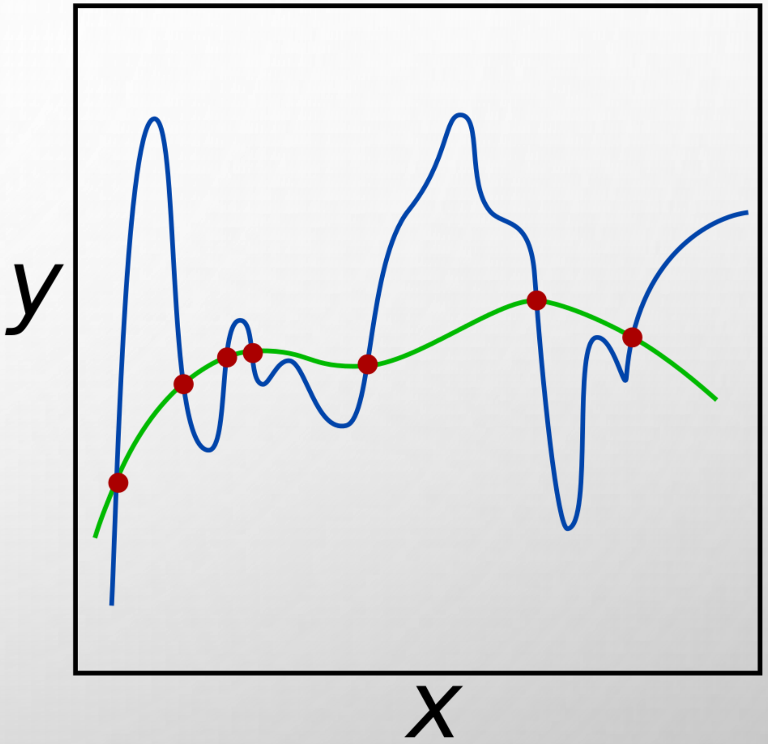
Although regularization procedures can be divided in many ways, the following delineation is particularly helpful:

- **Explicit regularization** is regularization whenever one explicitly adds a term to the optimization problem. These terms could be priors, penalties, or constraints. Explicit regularization is commonly employed with ill-posed optimization problems. The regularization term, or penalty, imposes a cost on the optimization function to make the optimal solution unique.
- **Implicit regularization** is all other forms of regularization. This includes, for example, early stopping, using a robust loss function, and discarding outliers. Implicit regularization is essentially ubiquitous in modern machine learning approaches, including stochastic gradient descent for training deep neural networks, and ensemble methods (such as random forests and gradient boosted trees).

In explicit regularization, independent of the problem or model, there is always a data term, that corresponds to a likelihood of the measurement and a regularization term that corresponds to a prior. By combining both using Bayesian statistics, one can compute a posterior, that includes both information sources and therefore stabilizes the estimation process. By trading off both objectives, one chooses to be more addictive to the data or to enforce generalization (to prevent overfitting). There is a whole research branch dealing with all possible regularizations. In practice, one usually tries a specific regularization and then figures out the probability density that corresponds to that regularization to justify the choice. It can also be physically motivated by common sense or intuition.

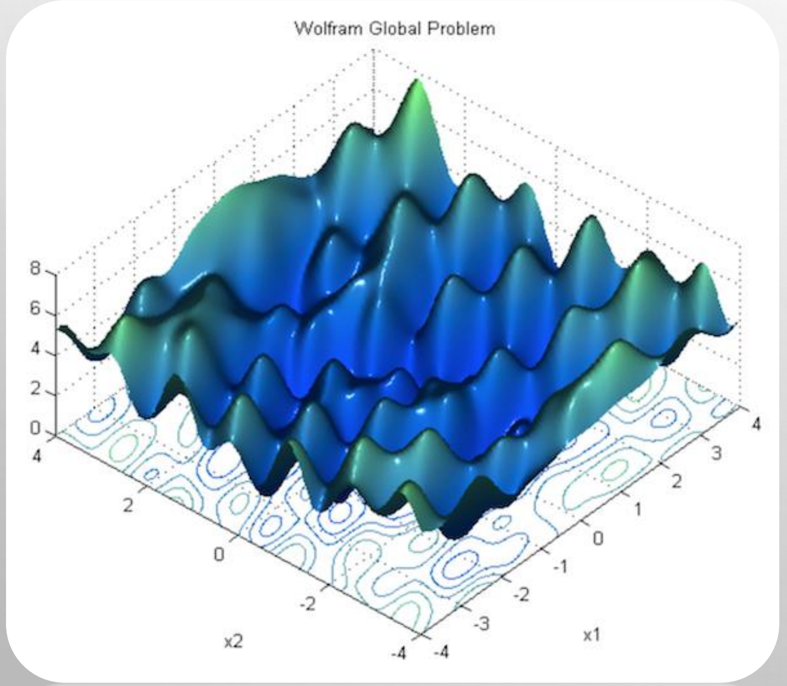
In machine learning, the data term corresponds to the training data and the regularization is either the choice of the model or modifications to the algorithm. It is always intended to reduce the generalization error, i.e. the error score with the trained model on the evaluation set and not the training data.[3]

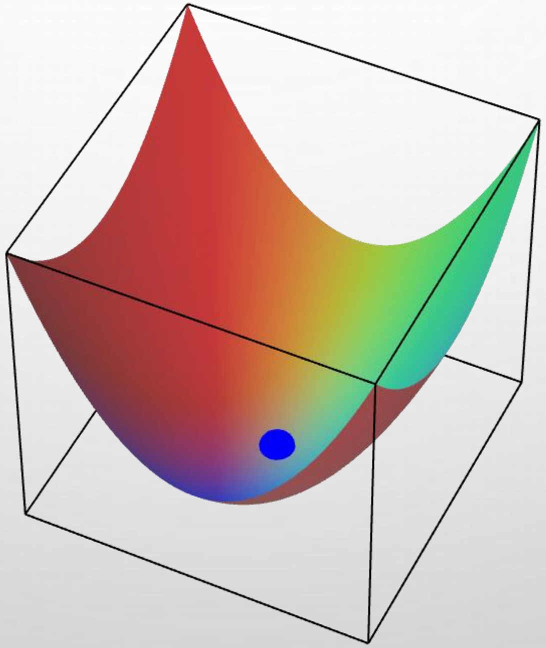
One of the earliest uses of regularization is Tikhonov regularization (ridge regression), related to the method of least squares.

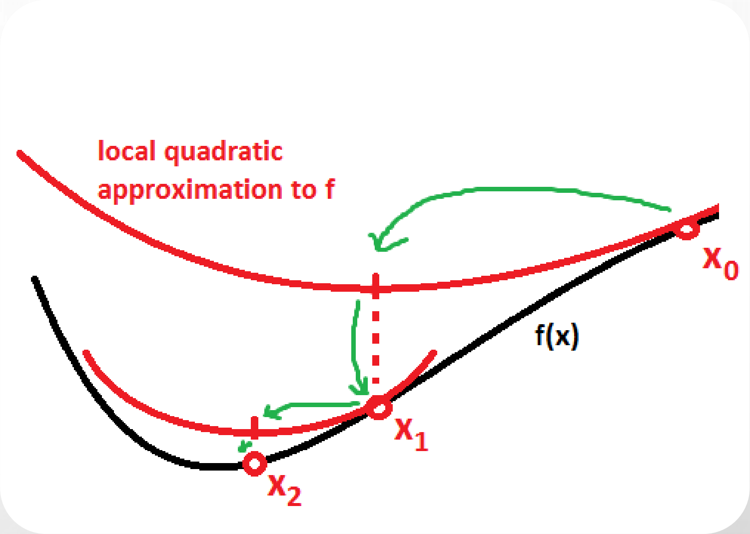


SUPERFÍCIE DO ERRO MÉDIO QUADRÁTICO

$$E = \frac{1}{N}\sum(y - f(w,x))^2$$







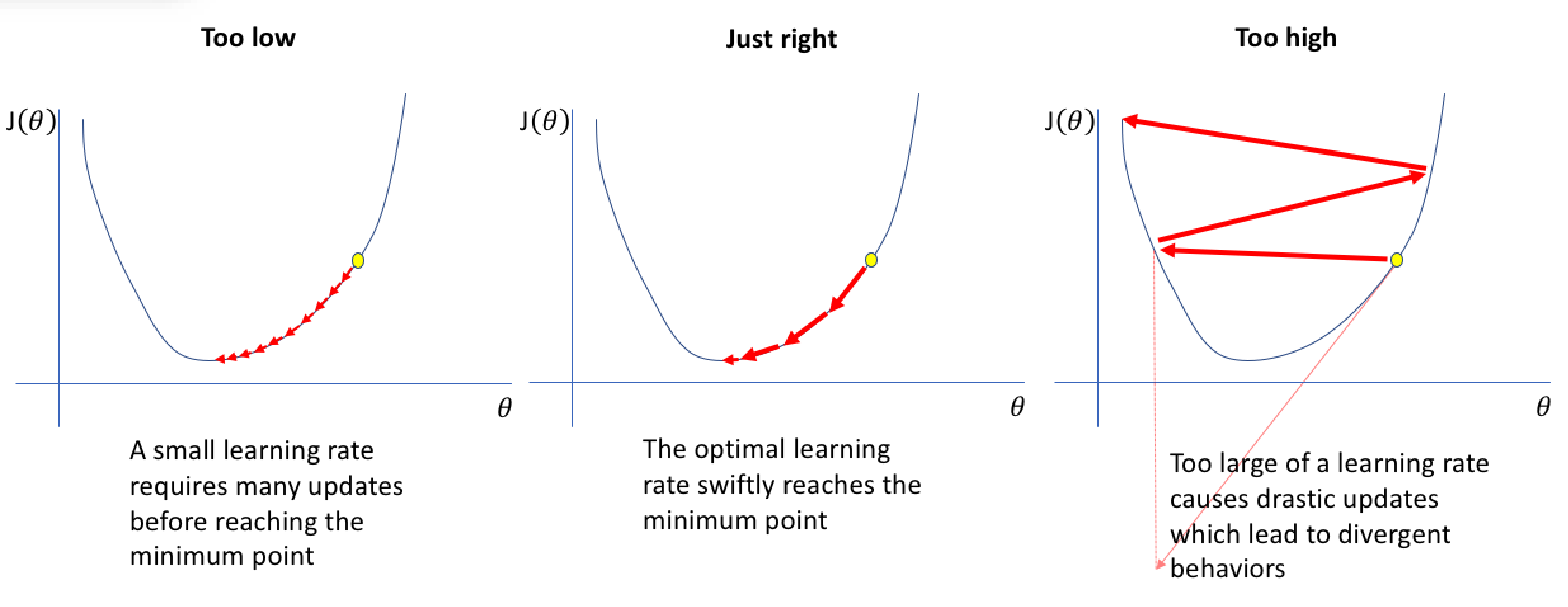
ALGORITMO DO GRADIENTE DESCENDENTE

$$\Delta w_{ij} = (\eta * \frac{\partial E}{\partial w_{ij}})$$

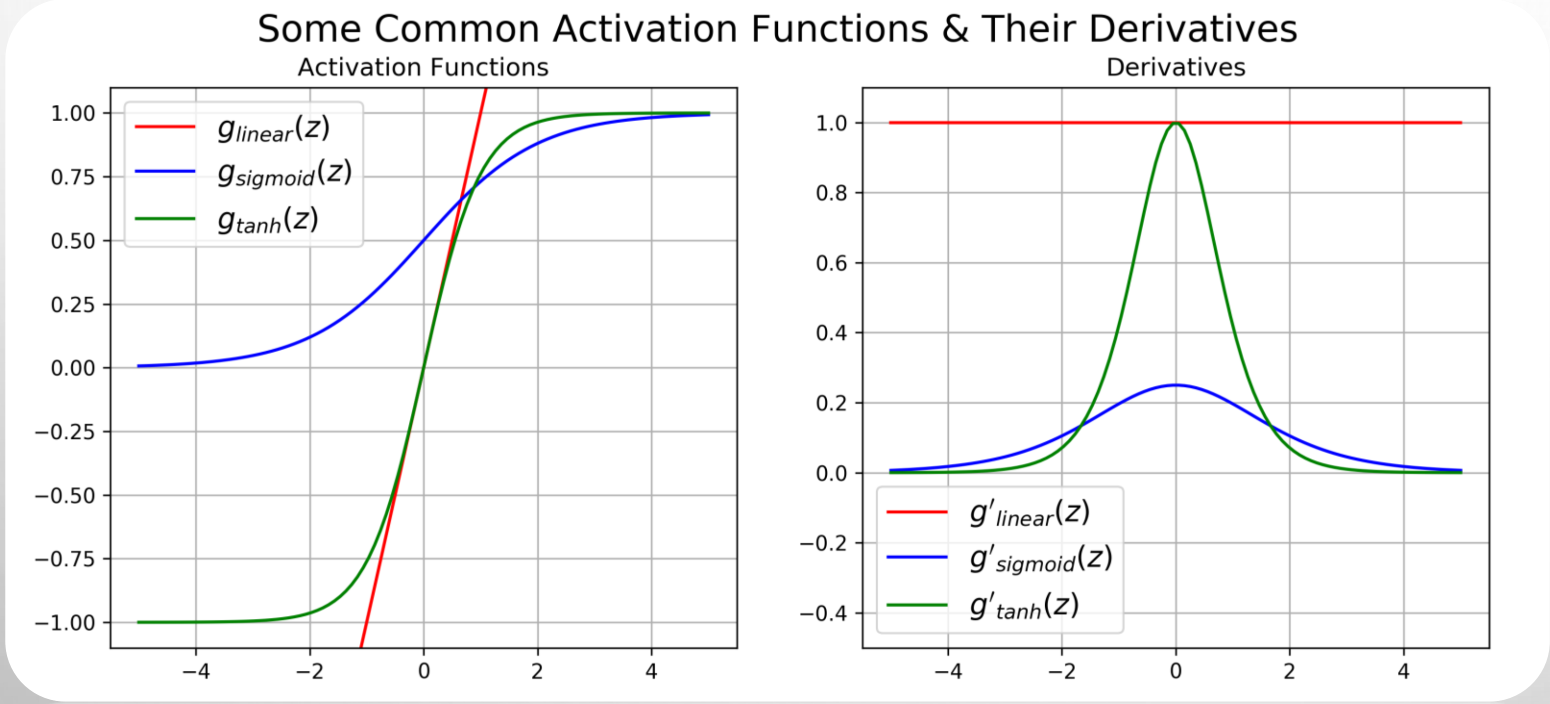
weight increment — learning rate — weight gradient

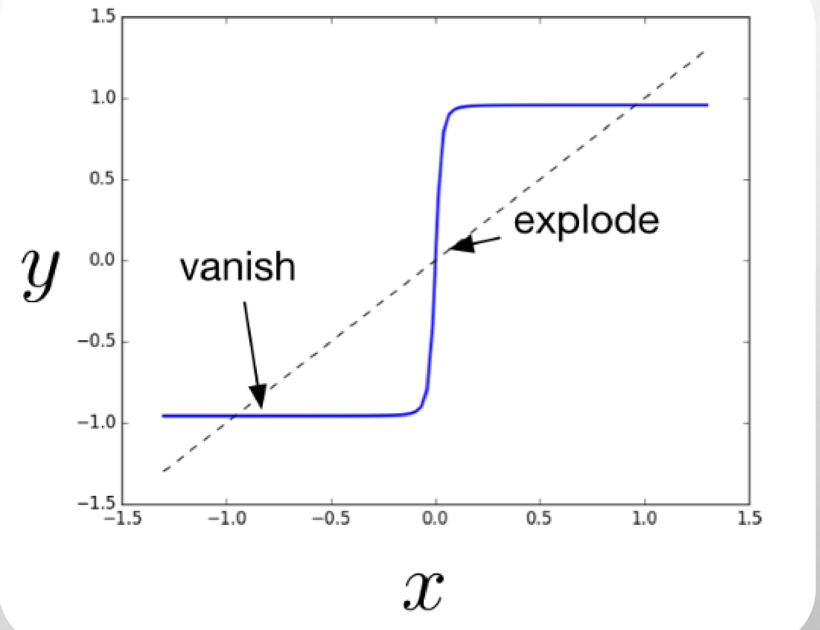
$$\Delta w_{ij} = (\eta * \frac{\partial E}{\partial w_{ij}}) + (\gamma * \Delta w_{ij}^{t-1})$$

momentum factor — weight increment, previous iteration



O PROBLEMA DA DISSIPAÇÃO DO GRADIENTE





OTIMIZADORES (REGULARIZADOS)

$$v_t^w = v_{t-1}^w + (\nabla w_t)^2$$
$$w_{t+1} = w_t - \frac{\eta}{\sqrt{v_t^w + \epsilon}} * \nabla w_t$$

$$v_t^b = v_{t-1}^b + (\nabla b_t)^2$$
$$b_{t+1} = b_t - \frac{\eta}{\sqrt{v_t^b + \epsilon}} * \nabla b_t$$

ADAGRAD

$$v_t^w = \beta * v_{t-1}^w + (1-\beta)(\nabla w_t)^2$$
$$w_{t+1} = w_t - \frac{\eta}{\sqrt{v_t^w + \epsilon}} * \nabla w_t$$

$$v_t^b = \beta * v_{t-1}^b + (1-\beta)(\nabla b_t)^2$$
$$b_{t+1} = b_t - \frac{\eta}{\sqrt{v_t^b + \epsilon}} * \nabla b_t$$

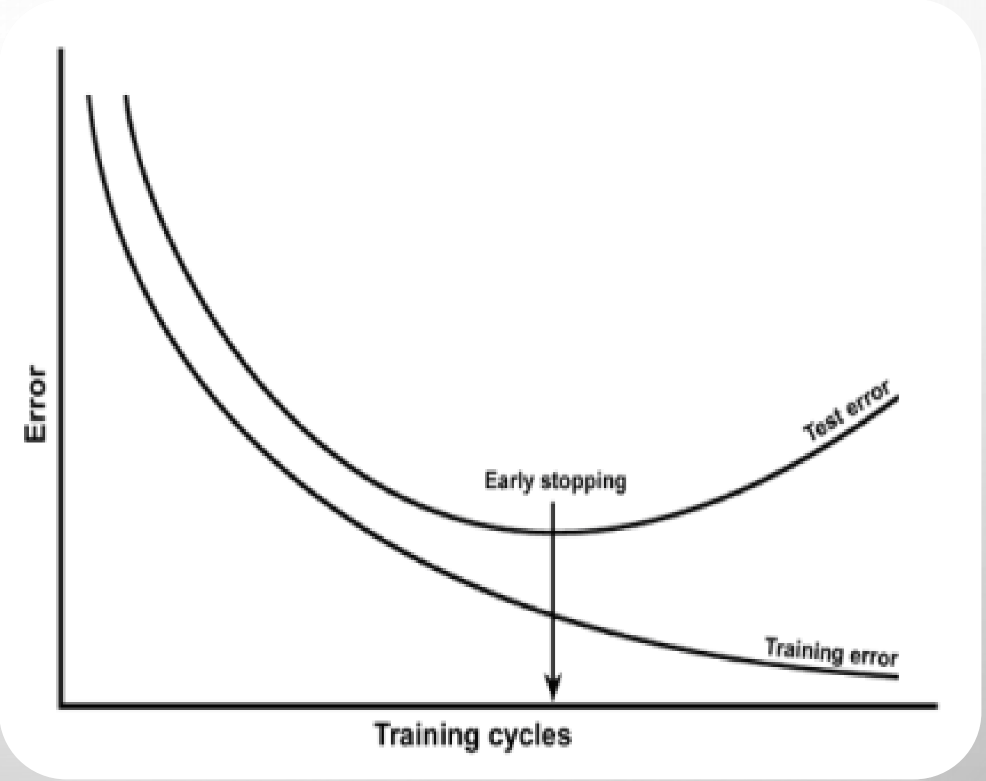
RMSPROP

$$m_t = \beta_1 * m_{t-1} + (1-\beta_1) * \nabla w_t$$
$$v_t = \beta_2 * v_{t-1} + (1-\beta_2) * (\nabla w_t)^2$$
$$\hat{m}_t = \frac{m_t}{1-\beta_1^t} \qquad \hat{v}_t = \frac{v_t}{1-\beta_2^t}$$
$$w_{t+1} = w_t - \frac{\eta}{\sqrt{\hat{v}_t + \epsilon}} * \hat{m}_t$$

ADAM

PARADA PREMATURA DO TREINAMENTO

- Aumento no Erro de Validação (Teste)
- Estabilidade da Figura de Mérito no Treino

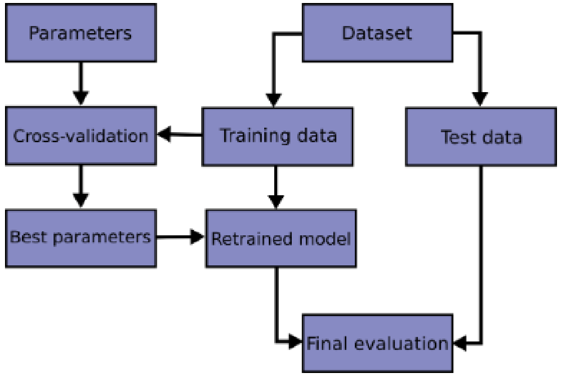


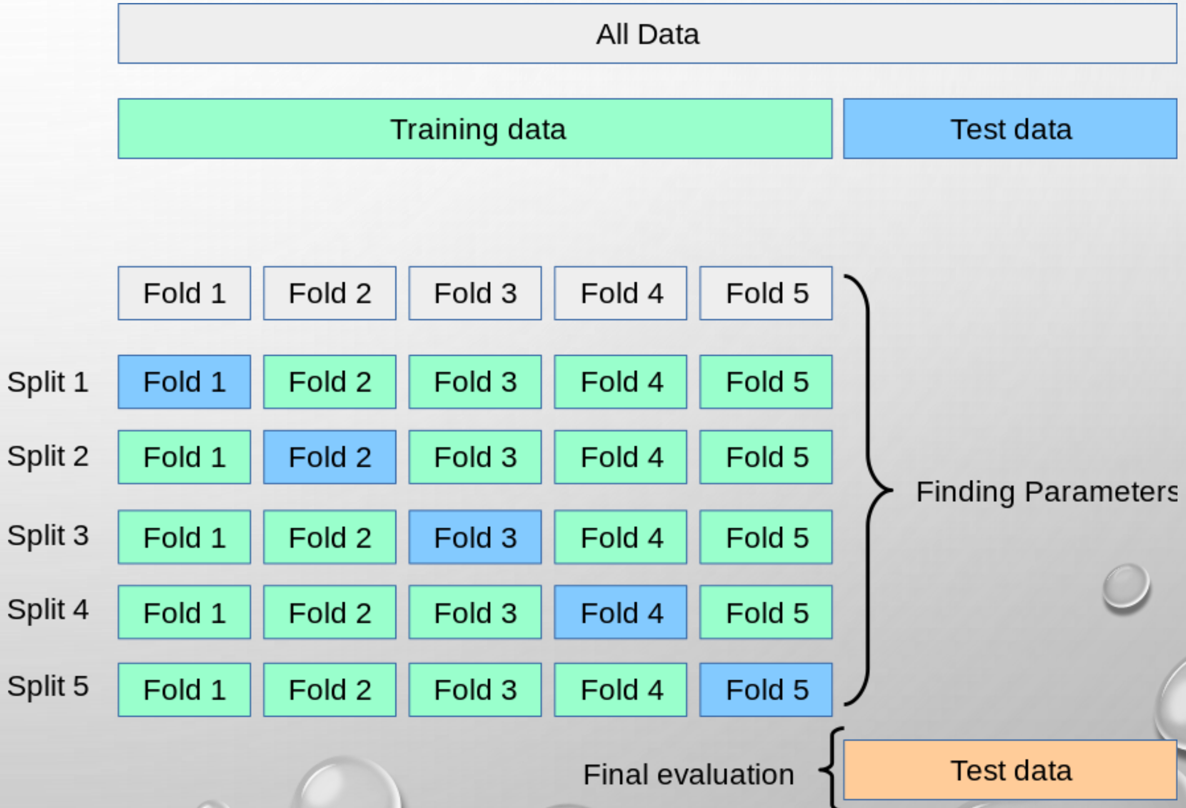
VALIDATION

VALIDAÇÃO CRUZADA

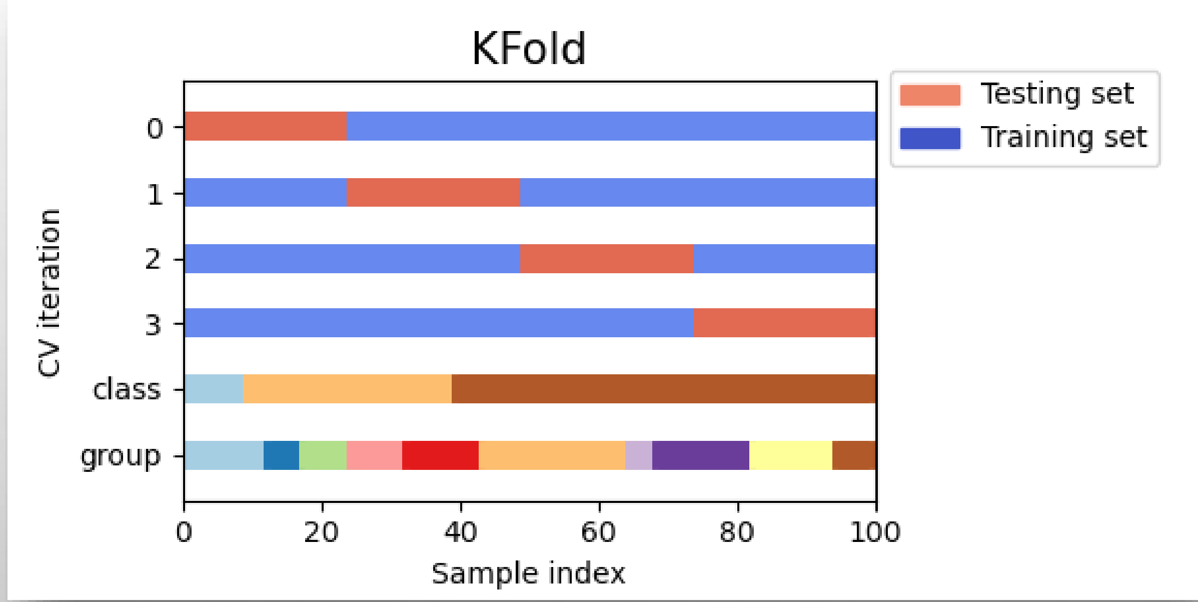
3.1. Cross-validation: evaluating estimator performance

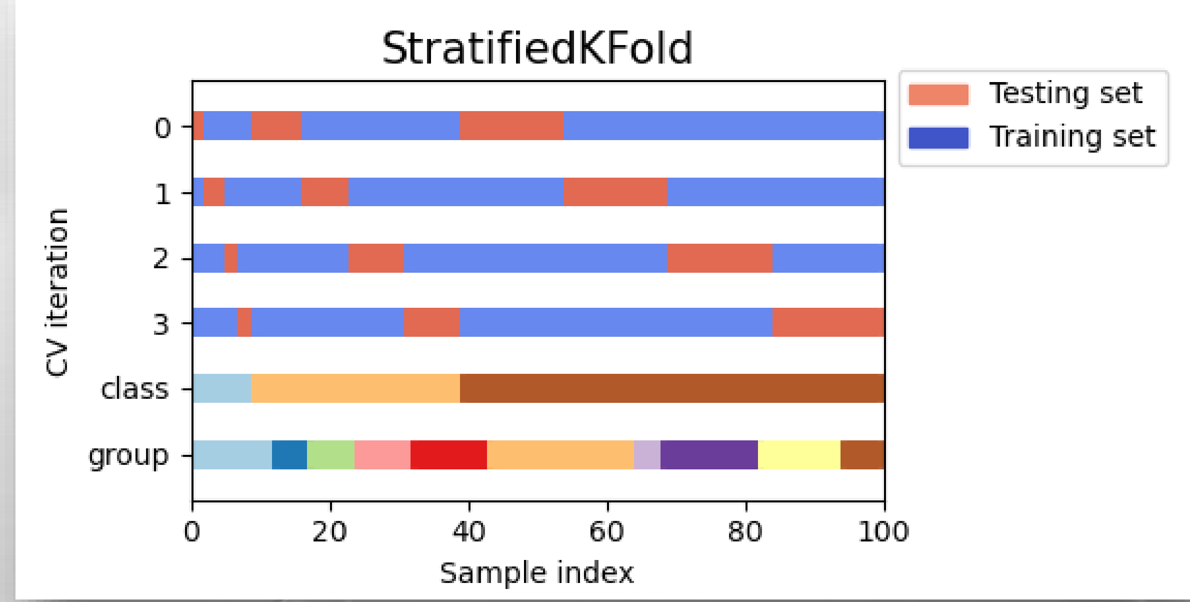
Learning the parameters of a prediction function and testing it on the same data is a methodological mistake: a model that would just repeat the labels of the samples that it has just seen would have a perfect score but would fail to predict anything useful on yet-unseen data. This situation is called **overfitting**. To avoid it, it is common practice when performing a (supervised) machine learning experiment to hold out part of the available data as a **test set** `X_test, y_test`. Note that the word "experiment" is not intended to denote academic use only, because even in commercial settings machine learning usually starts out experimentally. Here is a flowchart of typical cross validation workflow in model training. The best parameters can be determined by grid search techniques.





K-FOLDS & K-FOLDS ESTRATIFICADO





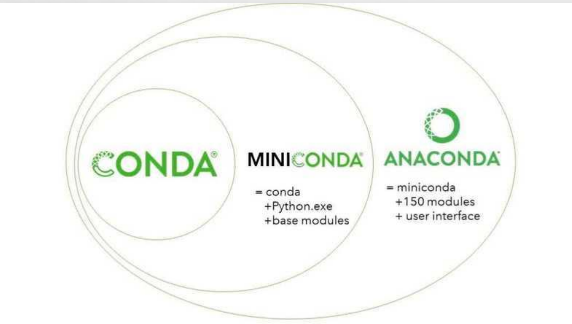
PARTE 2 : PRÁTICA

AMBIENTE PYTHON

5. Visualização

6. Machine Learning

4. Variáveis Aleatórias



1. Editor de Código

2. Gestor de Ambiente

3. Ambiente Python do Projeto

3. Notebook Dinâmico

PROBLEMA DE NEGÓCIO

Características das flores

Largura & comprimento da pétala

Largura & comprimento da sépala

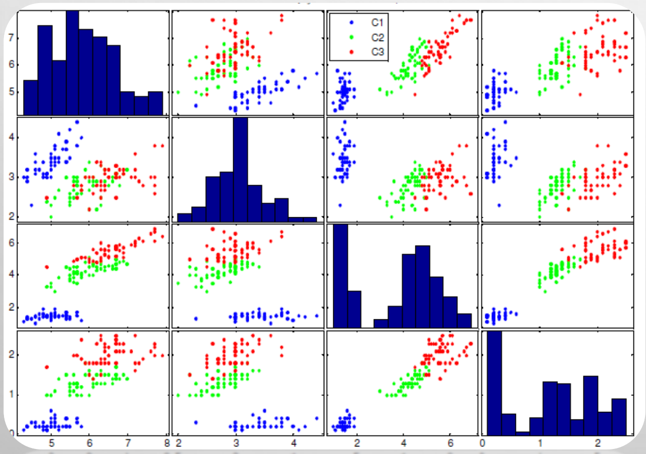
Iris Setosa

Iris Versicolor

Iris Virginica

REPRESENTAÇÃO

Iris Setosa | Iris Versicolor | Iris Virginica



http://archive.ics.uci.edu/ml/datasets/Iris

Características das flores

Largura & comprimento da pétala

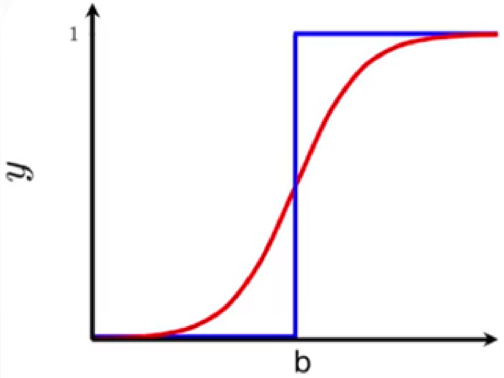
Largura & comprimento da sépala

Espaço de atributos com **4 dimensões!**

MODELAGEM

- REDE NEURAL FEED FORWARD
  - REPRESENTAÇÃO: 2 ATRIBUTOS
  - META-PARÂMETROS: 1..N NEURÔNIOS TANH NA CAMADA OCULTA
  - TREINAMENTO: BASE DE TREINO COMPLETA.
    - PRECISÃO / RECALL / ACURÁCIA
    - VALIDAÇÃO CRUZADA 10 FOLDS
  - ALGORITMO RMSPROP / ADAM
    - RMSPROP – TAXA DE APRENDIZADO FIXA “CAUTIOUS”
    - ADAM – TAXA DE APRENDIZADO COM DECAIMENTO “QUICKIE”

Iris Setosa | Iris Versicolor | Iris Virginica



$$y = \frac{1}{1+e^{-(w^T x + b)}}$$

- $\sum_{i=1}^{n} w_i x_i$

CLASSIFICADOR IRIS

EXERCÍCIO: REDE NEURAL FEEDFORWARD

PRÓXIMA AULA: CLASSIFICAÇÃO: TREINAMENTO ROBUSTO